

**CLAIMS**

**WHAT IS CLAIMED:**

- 1 1. A method for use in developing a program, comprising compiling at least a portion of  
2 a source code program defined by a waypoint during the editing of the source code program.
- 1 2. The method of claim 1, wherein compiling includes:  
2 identifying the waypoint in an edited source code during editing of the source code;  
3 and  
4 compiling the source code up to the identified waypoint before completing the edit of  
5 the source code.
- 1 3. The method of claim 1, wherein identifying the waypoint includes one of identifying  
2 the waypoint from a static definition and identifying the waypoint from a dynamic definition.
- 1 4. The method claim 1, further comprising:  
2 identifying a second waypoint in the source code during editing of the source code;  
3 and  
4 compiling the source code from the first waypoint to the second waypoint before  
5 completing editing of the source code.
- 1 5. The method of claim 1, further comprising:  
2 completing editing of the source code; and  
3 compiling the source code from the second waypoint to the end of the source code.
- 1 6. The method of claim 1, further comprising saving the edited source code.
- 1 7. The method of claim 1, further comprising compiling the source code from the  
2 waypoint to the end of the source code upon completing editing of the source code.
- 1 8. A method for use in developing a program, comprising:  
2 identifying a waypoint in an edited source code program during editing of the source  
3 code program; and  
4 compiling the source code program up to the identified waypoint before completing  
5 editing of the source code program.

1 9. The method of claim 8, wherein identifying the waypoint includes one of identifying  
2 the waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 10. The method claim 8, further comprising:  
2 identifying a second waypoint in the edited source code program during editing of the  
3 source code program; and  
4 compiling the source code program from the first waypoint to the second waypoint  
5 before completing editing of the source code program.

1 11. The method of claim 8, further comprising compiling the source code program from  
2 the waypoint to the end of the source code program upon completing editing of the source  
3 code program.

1 12. A method for modifying a compiler to engage in rapid compilation, comprising:  
2 identifying a file reader portion of the compiler; and  
3 modifying the identified file reader to read a portion of a source code program defined  
4 by a waypoint from a standard input open

1 13. The method of claim 12, wherein modifying the identified file reader to read from the  
2 standard input includes modifying the identified file reader to read from an open system call.

1 14. The method of claim 13, wherein modifying the identified file reader to read from the  
2 open system call includes modifying the identified file reader to read from a UNIX gcc  
3 command.

1 15. The method of claim 12, wherein the waypoint is identified by one of identifying the  
2 waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 16. The method of claim 12, wherein the waypoint defines a lower bound of the portion  
2 of the source code program.

1 17. The method of claim 12, wherein the waypoint defines an upper bound of the portion  
2 of the source code program.

1 18. A method for suspending compiler execution prior to reaching the end of a source  
2 code program, comprising:  
3 identifying a waypoint in the source code program;

4 compiling a portion of the source code program whose lower bound is defined by the  
5 identified waypoint; and  
6 suspending compilation of the source code program once the portion whose lower  
7 bound is identified by the waypoint is compiled.

1 19. The method of claim 18, wherein the waypoint is identified by one of identifying the  
2 waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 20. The method of claim 18, wherein suspending compilation of the source code program  
2 once the portion whose lower bound is identified by the waypoint is compiled includes at  
3 least one of removing a corresponding task from a work queue in an IDE, storing the  
4 compiled code in a shadow location, and suppressing errors or warning.

1 21. The method of claim 18, wherein the upper bound of the portion is defined by the  
2 start of the source code program or another waypoint.

1 22. A method for resuming compiler execution of a suspended compilation, comprising:  
2 triggering the compilation of a portion of a source code program whose upper bound  
3 is defined by an identified waypoint; and  
4 compiling the portion of the source code program whose upper bound is defined by  
5 the identified waypoint.

1 23. The method of claim 22, wherein triggering the compilation of the portion of the  
2 source code includes identifying the waypoint.

1 24. A method for identifying a command and associating it with a file that is being edited,  
2 comprising:  
3 modifying a file reader of a compiler to read from a standard input; and  
4 triggering the compilation of a portion of a source code program whose upper bound  
5 is defined by an identified waypoint;  
6 invoking the compiler to read the file from the modified file reader through the  
7 standard input.

1 25. The method of claim 24, wherein modifying the file reader to read from the standard  
2 input includes modifying the identified file reader to read from an open system call.

1 26. The method of claim 24, wherein modifying the file reader to read from the open  
2 system call includes modifying the identified file reader to read from a UNIX gcc command.

1 27. The method of claim 24, wherein triggering the compilation of the portion of the  
2 source code includes identifying the waypoint.

1 28. A method for building a source code program capable of suspending and resuming  
2 compilation, comprising:

3 identifying a waypoint in a source code program being edited;

4 triggering a compilation of a portion of the source code program defined by the  
5 waypoint;

6 compiling the portion of the source code program defined by the waypoint;

7 suspending the compilation of the portion defined by the waypoint once the  
8 compilation reaches the waypoint;

9 triggering the compilation of the remainder of the source code program; and

10 resuming the compilation of the source code program to compile the remainder.

1 29. The method of claim 28, wherein the waypoint is identified by one of identifying the  
2 waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 30. The method of claim 28, wherein triggering the compilation of the portion of the  
2 source code includes identifying the waypoint.

1 31. The method of claim 28, wherein suspending compilation of the source code program  
2 once the portion whose lower bound is identified by the waypoint is compiled includes at  
3 least one of removing a corresponding task from a work queue in an IDE, storing the  
4 compiled code in a shadow location, and suppressing errors or warning.

1 32. The method of claim 28, wherein the upper bound of the portion is defined by the  
2 start of the source code program or another waypoint.

1 33. The method of claim 28, wherein triggering the compilation of the remainder of the  
2 source code program includes identifying a second waypoint, saving the source code  
3 program, or ending an editing session.

1 34. A method for using a UNIX standard input read mechanism for speculative  
2 compilation of a source code program, comprising:

3 identifying a waypoint in an edited source code program during editing of the source  
4 code program; and

5 invoking a compile of at least a portion of a source code program defined by a  
6 waypoint during the editing of the source code program with a UNIX input  
7 read mechanism.

1 35. The method of claim 34, wherein the portion comprises a portion of the source code  
2 program defined by the start of the source code program and the waypoint.

1 36. The method of claim 34, wherein the portion comprises a portion of the source code  
2 program defined by the waypoint and the end of the source code program.

1 37. The method of claim 34, wherein the waypoint is identified by one of identifying the  
2 waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 38. A method for managing the output of a compile, comprising:  
2 compiling at least a portion of a source code program defined by a waypoint during  
3 the editing of the source code program in a first phase;  
4 compiling the remainder of the source code program in a subsequent phase; and  
5 notifying a user of any errors that may have occurred during the compilation.

1 39. The method of claim 38, wherein the portion comprises a portion of the source code  
2 program defined by the start of the source code program and the waypoint.

1 40. The method of claim 38, wherein the portion comprises a portion of the source code  
2 program defined by the waypoint and the end of the source code program.

1 41. The method of claim 38, wherein the waypoint is identified by one of identifying the  
2 waypoint from a static definition and identifying the waypoint from a dynamic definition.

1 42. The method of claim 38, further comprising scrapping the compiled first and second  
2 portions.

1 43. The method of claim 42, wherein scrapping the compiled first and second portions  
2 includes one of scrapping the compiled first and second portions responsive to the  
3 notification and scrapping the compiled first and second portions responsive to a user input.

1 44. A method for use in developing a program, comprising:  
2 identifying at least two or more instructions in a file to compile; and  
3 compiling the identified instructions while the file is being edited.

1 45. The method of claim 44, wherein the instructions are identified at a predetermined  
2 line number in the source code program, identifying the instructions at the point of insertion  
3 for a text editor, identifying the instructions after a predetermined number of branches as  
4 conditionals, identifying the instructions at a predetermined text offset.

1 46. The method claim 44, further comprising:  
2 identifying at least two more instructions in the file during editing; and  
3 compiling the second two or more instruction while the file is being edited.

1 47. The method of claim 44, further comprising:  
2 completing editing of the file; and  
3 compiling the remainder of the edited file.

1 48. The method of claim 44, further comprising saving the edited file.

1 49. The method of claim 44, further comprising compiling the remainder of the edited file  
2 upon completing editing of the file.

1 50. A method for compiling a source code program, comprising:  
2 identifying an upper bound for a portion of the source code program to compile;  
3 identifying a lower bound for the portion; and  
4 compiling the portion defined by the upper and lower bounds during an editing  
5 session on the source code program.

1 51. The method of claim 50, wherein at least one of identifying the upper bound and  
2 identifying the lower bound includes one of identifying the bound from a static definition and  
3 identifying the bound from a dynamic definition.

1 52. The method claim 50, further comprising:

2 identifying a third bound in the edited source code during editing of the source code;  
3 and  
4 compiling the source code from the lower bound to the third bound before completing  
5 editing of the source code.

1 53. The method of claim 50, further comprising compiling the source code from the lower  
2 bound to the end of the source code upon completing editing of the source code.